Submission Worksheet

CLICK TO GRADE

https://learn.ethereallab.app/assignment/IT114-003-F2024/it114-milestone-3-trivia-2024-m24/grade/vvh

Course: IT114-003-F2024
Assigment: [IT114] Milestone 3 Trivia 2024 M24
Student: Valeria C. (vvh)

Submissions:

Submission Selection
1 Submission [submitted] 12/3/2024 10:22:34 PM

Instructions

COLLAPSE

Implement the Milestone 3 features from the project's proposal document: https://docs.google.com/document/d/1h2aEWUoZ-etpz1CRI-StaWbZTjkd9BDMq0b6TXK4utI/view Make sure you add your ucid/date as code comments where code changes are done All code changes should reach the Milestone3 branch Create a pull request from Milestone3 to main and keep it open until you get the output PDF from this assignment. Gather the evidence of feature completion based on the below tasks. Once finished, get the output PDF and copy/move it to your repository folder on your local machine. Run the necessary git add, commit, and push steps to move it to GitHub Complete the pull request that was opened earlier Upload the same output PDF to Canvas

Branch name: Milestone3

Group

100%

Group: Basic UI
Tasks: 1
Points: 2

COLLAPSE

Task

100%

Group: Basic UI
Task #1: UI Panels
Weight: ~100%
Points: ~2.00

COLLAPSE

Details:

All code screenshots must include ucid/date.

App screenshots must have the UCID in the title bar like the lesson gave.

Columns: 4

Sub-Task 100% Group: Basic UI Task #1: UI Panels Sub Task #1: Show the

Sub-Task 100% Group: Basic UI Task #1: UI Panels Sub Task #2: Show the code

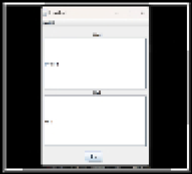
Sub-Task 100% Group: Basic UI Task #1: UI Panels Sub Task #3: Show the

Sub-Task 100% Group: Basic UI Task #1: UI Panels Sub Task #4: Show the code

Task Screenshots

Gallery Style: 2 Columns

4 2 1



connectionPanel showing up once app is running

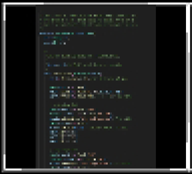
Caption(s) (required)✓

Caption Hint: *Describe/highlight what's being shown*

Task Screenshots

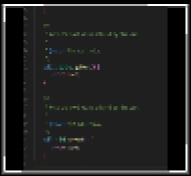
Gallery Style: 2 Columns

4 2 1



ConnectionPanel code allowing the user to input the host and port

second screenshot showing code related to the connectionpanel



third screenshot of code showing the connection panel

Caption(s) (required)✓

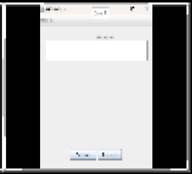
Caption Hint: *Describe/highlight what's being shown*

Task Response

Task Screenshots

Gallery Style: 2 Columns

4 2 1



UserDetailsPanel UI panel picture

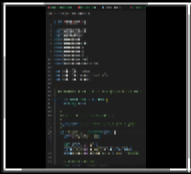
Caption(s) (required)✓

Caption Hint: *Describe/highlight what's being shown*

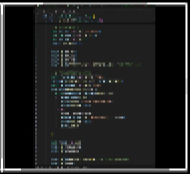
Task Screenshots

Gallery Style: 2 Columns

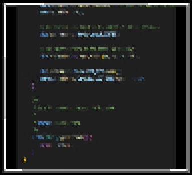
4 2 1



userdetailspanel code screenshot



second userdetailspanel code



3part userdetailspanel related code

Caption(s) (required)✓

Caption Hint: *Describe/highlight what's being shown*

Task Response Prompt

Briefly explain how it works and how it's used

Response:

Prompt

Briefly explain how it works and how it's used

Response:

The connectionpanel is where users input the host and port to connect to the server. When the program is running, this is the first panel that shows up indicating the host and port and also a "next" button, where also the program checks if the port is a valid number. If it's invalid, an error message is displayed. If the input is valid, the entered host and port are saved, and the program moves to the next screen.

The userdetailspanel class is designed for entering a username. It extends jpanel. The panel includes a text field for username input, an error label for validation messages, and two buttons "previous" for navigating back and "connect" for confirming the username. When the "connect" button is clicked, the entered username is validated to ensure it's not empty. If validation passes, the username is stored in the username field, logged using a logger utility, and a connection action is triggered. If the field is empty, an error message is displayed.

End of Task 1

End of Group: Basic UI
Task Status: 1/1

Group

100%

Group: Game Area
Tasks: 6
Points: 7

COLLAPSE

Task

100%

Group: Game Area
Task #1: ReadyCheck UI Panel
Weight: ~17%
Points: ~1.17

COLLAPSE

Details:

All code screenshots must include ucid/date.
App screenshots must have the UCID in the title bar like the lesson gave.

Columns: 2

Sub-Task

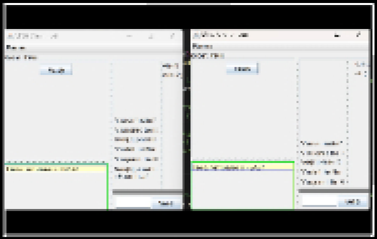
100%

Group: Game Area
Task #1: ReadyCheck UI Panel
Sub Task #1: Show the screen with the ready panel open in a fresh session

Task Screenshots

Gallery Style: 2 Columns

4 2 1



ready panel showing up once room is created

Caption(s) (required)✓
Caption Hint: *Describe/highlight what's being shown*

Sub-Task

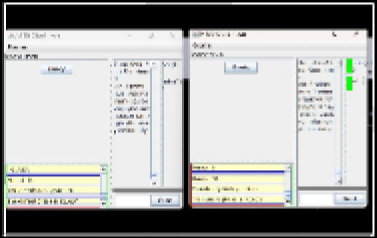
100%

Group: Game Area
Task #1: ReadyCheck UI Panel
Sub Task #2: Show the screen with the ready panel open after a session ends (there should be output in other parts of the UI showing this)

Task Screenshots

Gallery Style: 2 Columns

4 2 1



show the screen with the ready panel after the session ended

Caption(s) (required)✓
Caption Hint: *Describe/highlight what's being shown*

End of Task 1

Task

100%

Group: Game Area
Task #2: User List
Weight: ~17%
Points: ~1.17

COLLAPSE

Details:
All code screenshots must include ucid/date.

App screenshots must have the UCID in the title bar like the lesson gave.

Columns: 2

Sub-Task

100%

Group: Game Area
Task #2: User List
Sub Task #1: Show the UI indicating that a user locked in an answer for the round

Task Screenshots

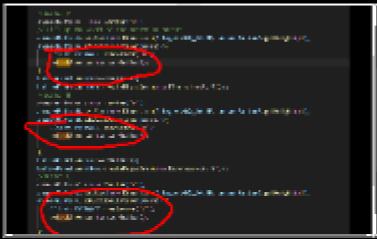
Sub-Task

100%

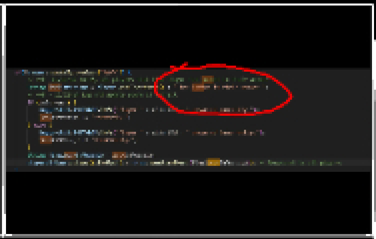
Group: Game Area
Task #2: User List
Sub Task #2: Show the related code (from server-side to UI) that marks the user list item properly

Gallery Style: 2 Columns

4 2 1



gamepanel.java code showing how lock answer button should display



code related of message displaying when player locked their answer.

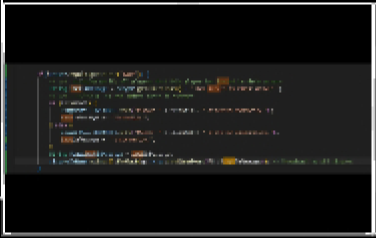
Caption(s) (required)✓

Caption Hint: *Describe/highlight what's being shown*

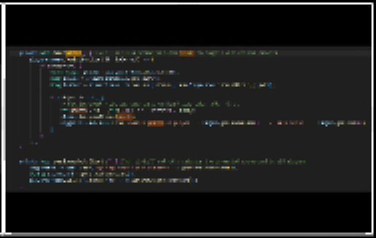
Task Screenshots

Gallery Style: 2 Columns

4 2 1



code related when user locks their answer to the server and gets correct/incorrect answer message



show awarded points to the players if answer is correct showing id and score across clients

Caption(s) (required)✓

Caption Hint: *Describe/highlight what's being shown*

Task Response Prompt

Explain in concise steps how this logically works

Response:

As intended for this code, the method should award points to players who answered correclty with a point system that considers the time taken to answer. the playerAnswers works as a map that associates each player, the cliendid, with wether their answer was correct. then the player retrieves the serverplayer object for the current clientid, the time the player took to answer, and the shortest answer time among all players. the base points are 10 and this are reduced based on how much slower the player was compared to the fastest response. this difference in time is divided by 1000 to convert the milliseconds to second and add the calculated points to the player's total scote, logs the awarded points and the player's new score it builds lock-in message that constructs a message starting with the player's name and the lock-in action and if the answer is correct, it logs that the player answered correctly showing "correctly" to lock-in message and the opposite if the answer is incorrect.

End of Task 2

Task

100%

Group: Game Area

Task #3: GameEventPanel

Weight: ~17%

Points: ~1.17

COLLAPSE

Details:

All code screenshots must include ucid/date.
App screenshots must have the UCID in the title bar like the lesson gave.

Columns: 2

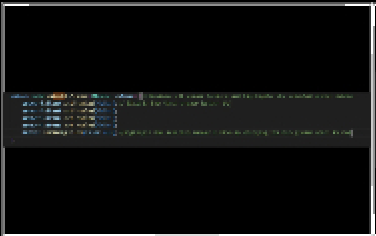
Sub-Task 100%

Group: Game Area
Task #3: GameEventPanel
Sub Task #1: Show the answer lock-in history

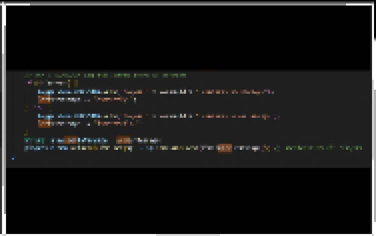
Task Screenshots

Gallery Style: 2 Columns

4 2 1



gamepanel.java code handle the lock-in logic when a player selects an answer during the game



gameroom.java code showing correct or wrong answer and broadcasting it to all players

Caption(s) (required)✓

Caption Hint: *Describe/highlight what's being shown*

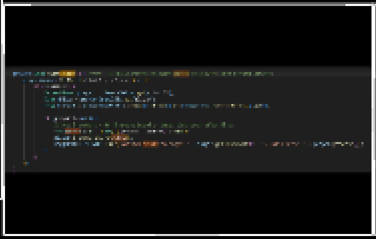
Sub-Task 100%

Group: Game Area
Task #3: GameEventPanel
Sub Task #2: Show earned points (should show the name and the points acquired)

Task Screenshots

Gallery Style: 2 Columns

4 2 1



gameroom.java showing the client name/id that awards the points to player and then showing total score/points for each round

scond part showing the at the end of round awardpoints gets called to display then players' score/points

Caption(s) (required)✓

Caption Hint: *Describe/highlight what's being shown*

Sub-Task 100%

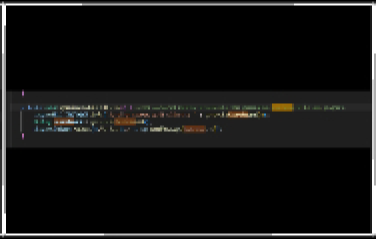
Group: Game Area
Task #3: GameEventPanel
Sub Task #3: Show the scoreboard updates from Milestone 2 (should display on the UI)

Task Screenshots

Gallery Style: 2 Columns

4 2 1

method to sync the current



method broadcasts

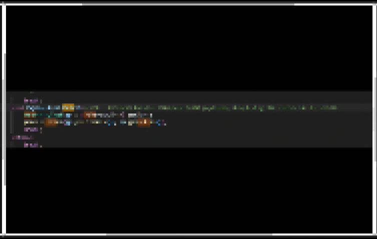
Sub-Task 100%

Group: Game Area
Task #3: GameEventPanel
Sub Task #4: Show the code for the UI flow (Client receiving to UI) for each of the 3 event examples above

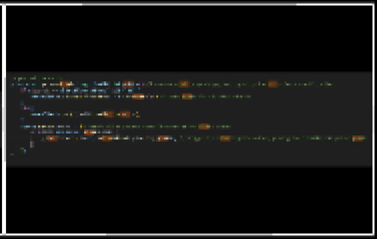
Task Screenshots

Gallery Style: 2 Columns

4 2 1



client side code case which is



triggers the onPointsUpdate

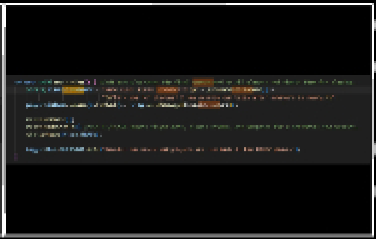
player scores

scoreboard to all players

supposed to handle the receive score playloads displaying the scores to the client

method to notify listeners of the updated points.

generate scoreboard where scoreboard gets formattedin descending order based on points showing score and client name



at the end of the sessions, the final scoreboard will show up along with the message to all players

Caption(s) (required)✓

Caption Hint: *Describe/highlight what's being shown*

Caption(s) (required)✓

Caption Hint: *Describe/highlight what's being shown*

Task Response Prompt

Explain in concise steps how this logically works

Response:

as intended with the code, it will process points payload recived from the server, updated the points for the specific player or resets the points for all the clients and updated the UI with the changes. when the payload is cast to pointspayload contains the clientID and points values and the processpoints method is called with these two values. for wach listener that implements the ipointsevent interface, it triggers the onpointsupdate method, passes the cliend id. therefore all player's points are reset to 0, and in the UI refreshes the scores

End of Task 3

Task

100%

Group: Game Area
Task #4: Question and Category
Weight: ~17%
Points: ~1.17

COLLAPSE

Details:

All code screenshots must include ucid/date.
App screenshots must have the UCID in the title bar like the lesson gave.

Columns: 3

Sub-Task

100%

Group: Game Area
Task #4: Question and Category
Sub Task #1: Show the question category

Task Screenshots

Gallery Style: 2 Columns

4 2 1

Sub-Task

100%

Group: Game Area
Task #4: Question and Category
Sub Task #2: Show the current question

Task Screenshots

Gallery Style: 2 Columns

4 2 1

Sub-Task

100%

Group: Game Area
Task #4: Question and Category
Sub Task #3: Show the UI code related this data (from Client receiving to

Task Screenshots

Gallery Style: 2 Columns

4 2 1

displays a question and its details: category, text, and answer options

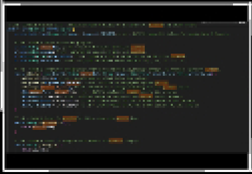
selects a category from the dropdown

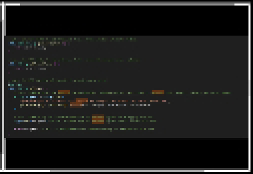
Caption(s) (required)✓
Caption Hint: *Describe/highlight what's being shown*

current question

displays question

Caption(s) (required)✓
Caption Hint: *Describe/highlight what's being shown*


QAPayload code related from question and answer


payload for handling trivia questions and answer options

Caption(s) (required)✓
Caption Hint: *Describe/highlight what's being shown*

Task Response Prompt

Explain in concise steps how this logically works

Response:

as intended with the code for the project, a dropdown category button should come up to allow users to select a category from available options, when the category is selected and confirmed, the selected category is sent to the server and the server responds with questions filtered by the selected category. the client should displays the questions dynamically in the UI. Clears any existing categories in the dropdown categoryDropdown. adds an "All" option (to view all categories. Iterates through the list of categories cats and adds each category as an option

End of Task 4

Task

100%

Group: Game Area
Task #5: Answers
Weight: ~17%
Points: ~1.17

COLLAPSE

Details:

All code screenshots must include ucid/date.
App screenshots must have the UCID in the title bar like the lesson gave.

Columns: 2

Sub-Task

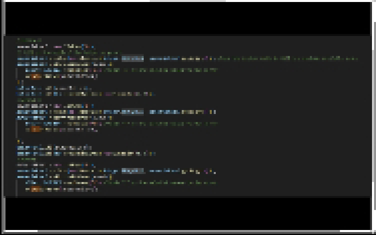
100%

Group: Game Area
Task #5: Answers
Sub Task #1: Show the current answers each with a button to lock in that choice (Locking in changes the color of the button and disables all answer choices)

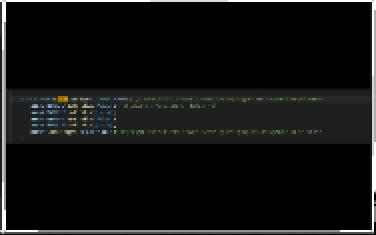
Task Screenshots

Gallery Style: 2 Columns

4 2 1



button represents an answer choice for the trivia questions



changes color of the button once answer is selected and disables answer choices

Caption(s) (required)✓

Caption Hint: *Describe/highlight what's being shown*

Sub-Task

100%

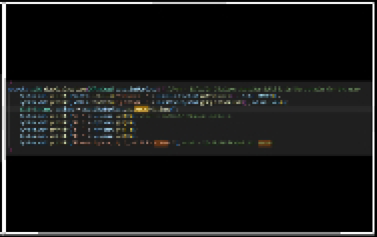
Group: Game Area
Task #5: Answers
Sub Task #2: Show the code related to managing and interacting with these components (UI to Client sending)

Task Screenshots

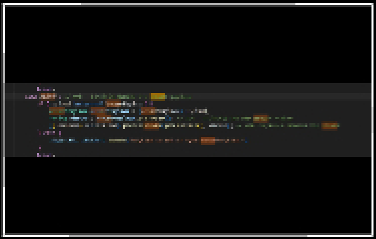
Gallery Style: 2 Columns

4 2 1

interaction of client with these components



options displaying for each question



handle the answer payload

Caption(s) (required)✓

Caption Hint: *Describe/highlight what's being shown*

Task Response Prompt

Explain in concise steps how this logically works

Response:

as intended with the code, the user should be able to see multiple answer buttons like a,b, c, d, on the screen and the user clicks one of these buttons to submit their answer. when the user click a button, an actionlistener associated with the button is triggered, where the button's actionlistener sends the selected answer to the server. it calls a method onlockinanswer to visually lock in the user's choice. the selected answer is sent to the server for the validation and scoring via the client.instance.sendanswer method and on the lockinanswer method, it disables all answer buttons, highlights the selected button changes its background highlights the selected button changes its background color to red to indicate the locked-in choice

End of Task 5

Task

100%

Group: Game Area
Task #6: Countdown Timer UI
Weight: ~17%
Points: ~1.17

COLLAPSE

Details:

All code screenshots must include ucid/date.
App screenshots must have the UCID in the title bar like the lesson gave.

Columns: 2

Sub-Task

100%

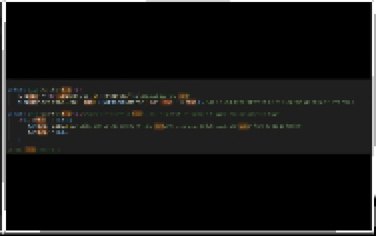
Group: Game Area
Task #6: Countdown Timer UI
Sub Task #1: Show the UI of the countdown (few examples to show it changes)

Task Screenshots

Gallery Style: 2 Columns

4 2 1

time handlers in gameroom.java



turn-based timer in the game

handle timer and user interaction when turn ends or starts

Caption(s) (required)✓

Sub-Task

100%

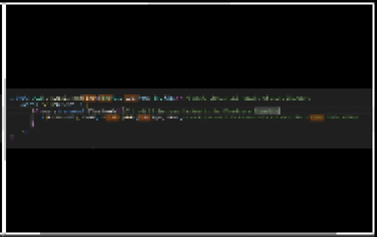
Group: Game Area
Task #6: Countdown Timer UI
Sub Task #2: Show the code related to managing the timer

Task Screenshots

Gallery Style: 2 Columns

4 2 1

Handles real-time update for timer

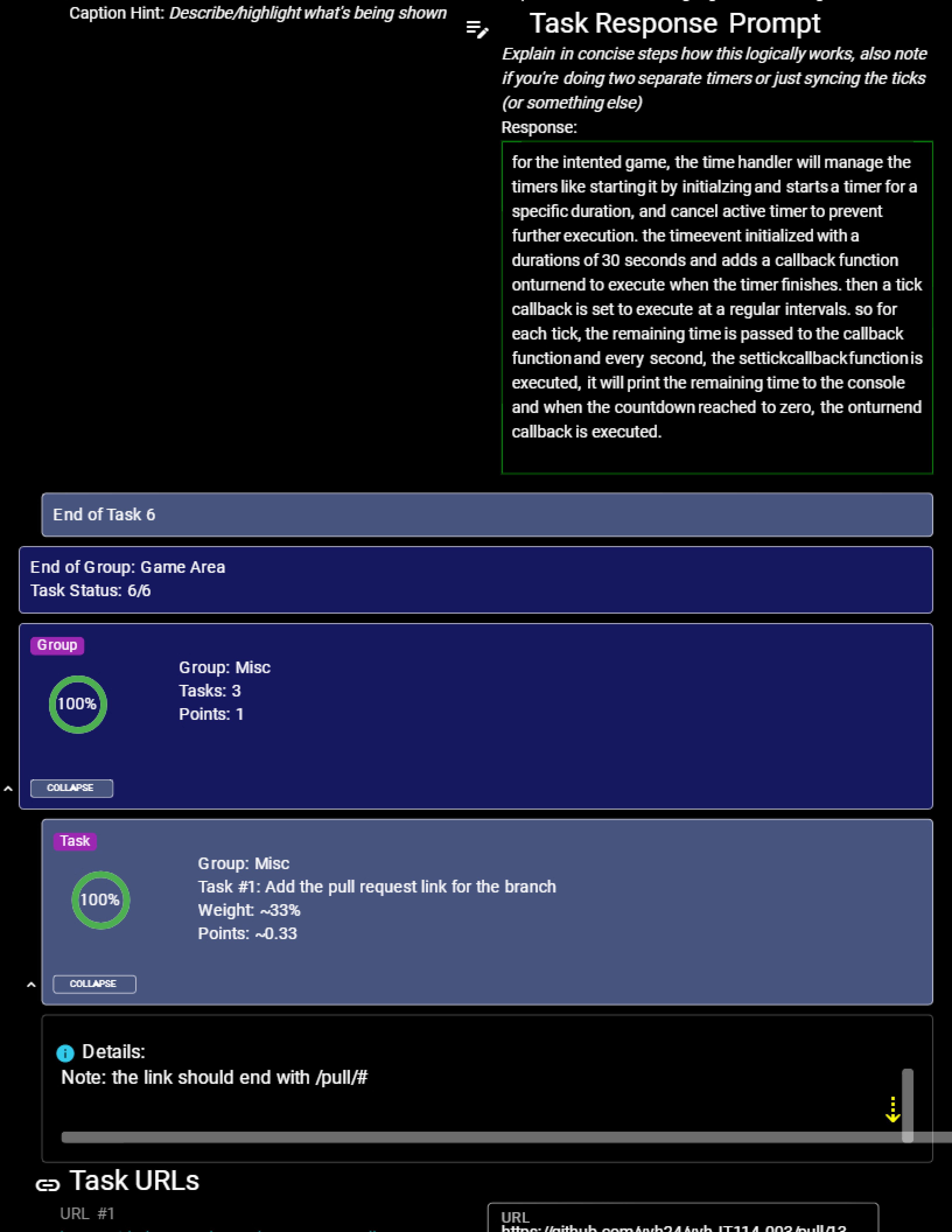


handle the timer update

onTimerUpdate method is a callback triggered when a timer update event occurs

Caption(s) (required)✓

Caption Hint: *Describe/highlight what's being shown*



Caption Hint: *Describe/highlight what's being shown*

Task Response Prompt

Explain in concise steps how this logically works, also note if you're doing two separate timers or just syncing the ticks (or something else)

Response:

for the intented game, the time handler will manage the timers like starting it by initialzing and starts a timer for a specific duration, and cancel active timer to prevent further execution. the timeevent initialized with a durations of 30 seconds and adds a callback function onturnend to execute when the timer finishes. then a tick callback is set to execute at a regular intervals. so for each tick, the remaining time is passed to the callback function and every second, the settickcallback function is executed, it will print the remaining time to the console and when the countdown reached to zero, the onturnend callback is executed.

End of Task 6

End of Group: Game Area
Task Status: 6/6

Group: Misc
Tasks: 3
Points: 1

Group: Misc
Task #1: Add the pull request link for the branch
Weight: ~33%
Points: ~0.33

Details:
Note: the link should end with /pull/#

Task URLs

URL #1

URL
https://github.com/wh24/wh_IT114-003/pull/13

https://github.com/vvh24/vvh-IT114-003/pull/13 https://github.com/vvh24/vvh-IT114-003/pull/13

End of Task 1

Task

100%

Group: Misc
Task #2: Talk about any issues or learnings during this assignment
Weight: ~33%
Points: ~0.33

COLLAPSE

Task Response Prompt

Response:

I am not going to lie. Milestone 3 made me one of the hardest time in probably my whole school life. I had such a hard time finding a way for the game to work. I appreciate help providing us with a template for an easy integration which at least fo the first 2 points of Milestone 3, it was already done and easy. But the game part was extremely difficult for me because I honestly had no clue of how to make it work because first my questions were not showing up and then I tried to make these buttons, but I dont think I accomplish the expectations in my mind to it. So far, I tried to combine milestone 3 and 4 so it is easier for me since everything is related, but I still need to make some fixes because the program is running, but as said before, questions are not displaying neither the categories, so for milestone 4, I have to make some fixes and adjustments to it

End of Task 2

Task

100%

Group: Misc
Task #3: WakaTime Screenshot
Weight: ~33%
Points: ~0.33

COLLAPSE

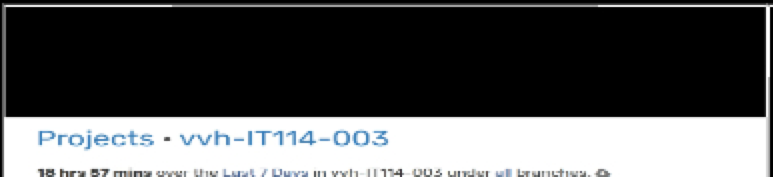
Details:

Grab a snippet showing the approximate time involved that clearly shows your repository. The duration isn't considered for grading, but there should be some time involved

Task Screenshots

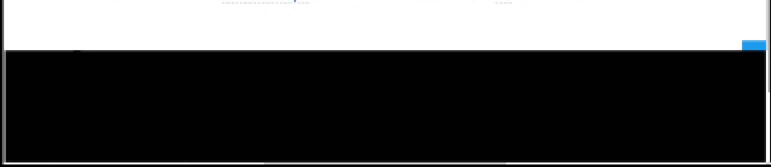
Gallery Style: 2 Columns

4 2 1

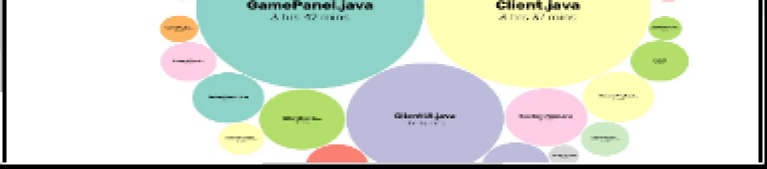


Projects • vvh-IT114-003

18 hrs 57 mins over the Last 7 Days in vvh-IT114-003 under all branches.



wakatime part1



wakatime part2

wakatime part3

End of Task 3

End of Group: Misc
Task Status: 3/3

End of Assignment